Parameterized Tractability of the Maximum-Duo Preservation String Mapping Problem

Stefano Beretta[a,b,*], Mauro Castelli[c], Riccardo Dondi[d]

[a]*Istituto di Tecnologie Biomediche, Consiglio Nazionale delle Ricerche, Segrate - Italia*
[b]*Dipartimento di Informatica, Sistemistica e Comunicazione, Università degli Studi di Milano - Bicocca, Milano - Italia*
[c]*NOVA IMS, Universidade Nova de Lisboa, Lisboa - Portugal*
[d]*Dipartimento di Lettere, Filosofia e Comunicazione, Università degli Studi di Bergamo, Bergamo - Italia*

Abstract

In this paper we investigate the parameterized complexity of the Maximum-Duo Preservation String Mapping Problem, the complementary of the Minimum Common String Partition Problem. We show that this problem is fixed-parameter tractable when parameterized by the number k of conserved duos, by first giving a parameterized algorithm based on the color-coding technique and then presenting a reduction to a kernel of size $O(k^6)$.

Keywords: Computational Biology, Common String Partition, Parameterized Algorithms, Kernelization

1. Introduction

Minimum Common String Partition (MCSP) is a problem that emerged in the field of comparative genomics [9] and, in particular, in the context of ortholog gene assignments [9]. Given two strings (genomes) A and B of length n, MCSP asks for a partition of the two strings into a minimum cardinality multiset of identical substrings. The complexity of this problem has been previously studied in the literature. More precisely, the MCSP problem is known to be APX-hard, even when each symbol has at most 2 occurrences in each input string [15], while it admits a polynomial-time algorithm when each symbol occurs exactly once in each input string. Approximation algorithms for this problem have been proposed in [10, 11, 15, 18]. More precisely, an $O(\log n \log^* n)$-approximation algorithm has been given in [11], while an $O(k)$-approximation algorithm, when the number of occurrences of each symbol is bounded by k, has been given in [18]. When $k = 2$ and $k = 3$ respectively, approximation algorithms of factor 1.1037 and 4, respectively, have been given in [15].

The parameterized complexity [13, 19] of MCSP has also been investigated. First, fixed-parameter algorithms have been given when the problem is parameterized by two parameters. In [12] this problem has been shown to be fixed-parameter tractable, when parameterized by the number of substrings in the solution and by the repetition number of the input strings. Then, the MCSP problem has been shown to be fixed-parameter tractable when parameterized by the number of substrings in the solution and the maximum number of occurrences of a symbol in an input string [6, 17]. Recently, MCSP has been shown to be fixed-parameter tractable when parameterized by the single parameter number of substrings of the partition [7].

Here, we consider the complementary of the MCSP problem, called Maximum-Duo Preservation String Mapping Problem, where instead of minimizing the number of identical substrings in the partition, we aim to maximize the number of preserved duos. Informally, the idea is to define a mapping between the positions

[*] Corresponding author

Email addresses: stefano.beretta@disco.unimib.it (Stefano Beretta), mcastelli@novaims.unl.pt (Mauro Castelli), riccardo.dondi@unibg.it (Riccardo Dondi) (having a same symbol) of the input strings such that the number of adjacent positions of one string mapped to adjacent positions of the other one (called preserved duos) is maximized. This problem has been proposed in [8], has been shown to be APX-hard when each symbol has at most 2 occurrences in each input string [4], and can be approximated within factor $\frac{1}{4}$ [4].

In this work, we study the parameterized complexity of the Maximum-Duo Preservation String Mapping Problem, where the parameter is the number of preserved duos. More precisely, after introducing definitions and properties of Maximum-Duo Preservation String Mapping in Section 2, we describe in Section 3 a fixed-parameter algorithm for the problem, based on the color-coding technique. Then, in Section 4, we present a reduction to a polynomial kernel of size $O(k^6)$ obtained by appropriately selecting subsets of duos of the input strings and by defining two new strings containing these subsets on an extended alphabet.

The results described in this paper are mainly of theoretical interest, since a solution of the Maximum-Duo Preservation String Mapping Problem is expected to preserve many adjacencies. Indeed, the complexity of the first algorithm and size of the kernel both depend (exponentially and polynomially, respectively) on the number of preserved adjacencies. However, the fixed-parameter algorithms we propose can be of interest for describing the parameterized complexity status of the combinatorial problems related to string partitioning. For example, while it is still unknown whether Minimum Common String Partition Problem admits a polynomial kernel, the result in Section 4 shows that such a kernel exists for the complementary problem.

2. Preliminaries

In this section, we introduce some concepts that will be used in the rest of the paper and we give the formal definition of the Maximum-Duo Preservation String Mapping Problem. Fig. 1 illustrates some of the definitions we give in this section.

Let Σ be a non-empty finite set of symbols. Given a string A over Σ, we denote by $|A|$ the length of A and by $A[i]$, with $1 \leq i \leq |A|$, the symbol of A at position i. Moreover, we denote by $A[i, j]$, with $1 \leq i < j \leq |A|$, the substring of A starting at position i and ending at position j. Given a string A, a *duo* is an ordered pair of consecutive elements $(A[i], A[i+1])$. Consider a duo $(A[i], A[i+1])$ in a string A; it is *preservable* if there exists a duo $(B[j], B[j+1])$ in a string B such that $A[i] = B[j]$ and $A[i+1] = B[j+1]$. Symmetrically, the property holds for a duo of string B.

Given two strings A and B, such that B is a permutation of A, we say that A and B are *related*. In the rest of the paper we assume that $|A| = |B| = n$.

Given two related strings A and B, a *mapping* m of A into B is a bijective function from the positions of A to the positions of B such that $m(i) = j$ implies that $A[i] = B[j]$, i.e. the two positions i, j of the two strings contain the same symbol. A *partial mapping* m of A into B is a bijective function from a subset of positions of A to a subset of positions of B such that $m(i) = j$ implies that $A[i] = B[j]$. The definition of mapping and partial mapping can be extended to two sets of duos of related strings A and B, that is if positions i and $i+1$, with $1 \leq i \leq n-1$, are mapped into positions j and $j+1$, with $1 \leq j \leq n-1$, we say that duo $(A[i], A[i+1])$ is mapped into duo $(B[j], B[j+1])$.

Given two related strings A and B, and a mapping m of the positions of A into the positions of B, a duo $(A[i], A[i+1])$ is preserved if $m(i) = j$ and $m(i+1) = j+1$ (see Figure 1 for an example).

Now, we give the definition of the Maximum-Duo Preservation String Mapping Problem (in its decision version).

Maximum-Duo Preservation String Mapping Problem (Max-Duo PSM)
Input: two related strings A and B, an integer k.
Output: is there a mapping m of A into B such that the number of preserved duos is at least k?

In this paper, we focus on the parameterized complexity of Max-Duo PSM, when parameterized by the number k of preserved duos.

Consider a string S, with $S \in \{A, B\}$, and a string $\bar{S} \in \{A, B\} \setminus \{S\}$. Given two positions $1 \leq i < j \leq n$, we denote by $d_{S(i,j)}$ the sequence of *consecutive duos* $(S[i], S[i+1])$, ..., $(S[j-1], S[j])$; the length of $d_{S(i,j)}$

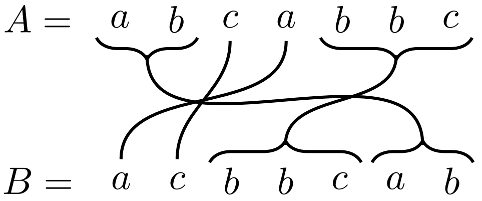


Figure 1: An example of two related strings A and B. The mapping of their positions is represented by connecting positions/substrings. Position 1 and 2 of A are mapped into positions 6 and 7 of B, hence duo $(A[1], A[2])$ of A is preserved; position 1 in A induces duo $(A[1], A[2])$. Similarly, the sequence $d_{A(5,7)}$ of consecutive duos is mapped into the sequence $d_{B(3,5)}$ of consecutive duos; hence duos $(A[5], A[6])$, $(A[6], A[7])$ of A are preserved. The number of preserved duos induced by the mapping is 3.

is the number $j - i$ of consecutive duos in it. Given the sequence $d_{S(i,j)}$ of consecutive duos, the *string corresponding* to $d_{S(i,j)}$ is $S[i, j]$. Given a string $S[i, j]$ the sequence of consecutive duos *induced* by $S[i, j]$ is $d_{S(i,j)}$. We say that position i, with $1 \leq i \leq n - 1$, of a string S, induces duo $(S[i], S[i + 1])$.

We denote by $d_{S(i-k,i+k)}$ the sequence of duos between position $S[l]$, where $l = \max\{1, i - k\}$, and position $S[r]$, where $r = \min\{n, i + k\}$.

Parameterized Complexity

We briefly overview the main concepts about parameterized complexity that will be useful in the rest of paper. We refer the reader to [13, 19] for an introduction to parameterized complexity.

A decision problem Π is fixed-parameter tractable under a parameter k when there exists an algorithm of time complexity $O(f(k)\text{poly-}time(q))$, which computes whether Π on instance (I, k) is decidable, where q is the size of an instance of problem Π, and $f(k)$ is a computable function that depends only on k (and not on q).

A reduction to a kernel for a given parameterized problem Π parameterized by k is a polynomial-time algorithm that, starting from an instance (I, k) (where k is the parameter) of Π, computes an instance (I', k') (called *kernel*) such that $k' \leq k$ and the size of I' is a function on k, and such that Π on instance (I, k) admits a solution if and only if Π on instance (I', k') admits a solution. It is well-known [13, 19] that any problem which is fixed-parameter tractable can be reduced to a kernel of exponential size. Moreover, there exist problems that admit a polynomial-size kernel, that is the size of I' is a polynomial function in k.

Our first FPT-algorithm is based on the well-known *color-coding* technique, introduced in [1]. Our color-coding approach is based on the definition of a perfect family of hash functions, used in [1] to derandomize the technique. Color-coding is a technique widely used to design fixed-parameter algorithms. While most of the applications of this technique are for graph problems [1, 14], it has been recently applied to problems on strings [2, 3, 5].

Now, we give the formal definition of perfect families of hash functions, on which the color-coding technique is based.

Definition 2.1. Let U be a set of cardinality n and let C be a set of colors having size k. A family F of hash functions from U to C is called *perfect* if for any subset $W \subseteq U$, such that $|W| = k$, there exists a function $f \in F$ such that for each $x, y \in W$, $f(x) \neq f(y)$.

Moreover, a perfect family F of hash functions from U to $C = \{c_1, \dots, c_k\}$, having size $O(\log |U| 2^{O(k)})$, can be constructed in time $O(2^{O(k)}|U| \log |U|)$ (see [1]).

We give an informal description of the color-coding technique. Consider a problem Π that, given a set U of n elements, aims to identify whether there exists a feasible solution which is a subset of U of size k ($k < n$). The existence of such a subset can be decided by enumerating all the subsets of size k in time $O(n^k)$. However, a family F of perfect hash functions can be used for some combinatorial problems to decide if such a subset exists or not in time $f(k)\text{poly-}time(n)$, by first using a function $f \in F$ that associates a distinct color to each element of a subset of size k and then applying dynamic programming.

3. An FPT Algorithm

In this section, we present an FPT algorithm for Max-Duo PSM parameterized by the number of preserved duos k between the input strings A and B. Notice that k preserved duos are induced by k positions as proved in the following lemma.

Lemma 1. *Consider two related strings A and B, a solution of Max-Duo PSM on the instance (A, B, k), and consider the p positions of B that induce k preserved duos. Then it holds $p = k$.*

Proof. The lemma follows easily from the fact that each preserved duo $(S[i], S[i+1])$ of a string S, with $S \in \{A, B\}$, is induced by position i of S. □

Thanks to Lemma 1, we present an FPT algorithm which is parameterized by the number of positions of B that induce k preserved duos.

Given an integer k, let $C = \{c_1, \ldots, c_k\}$ be a set of k *colors*. Let F be a family of perfect hash functions from the positions of B to the set C. Informally, we assign k distinct colors to the positions of B that may induce preserved duos, and by dynamic programming we compute if there exist k distinct positions in A that are mapped to these candidate duos of B. By Def. 2.1, we consider a function $f \in F$ that associates a distinct color to each of the k positions of B that induces a preserved duo.

Define $D[i, C']$, for $1 \leq i \leq n$, and $C' \subseteq C$, as a function equal to 1 if there exist a set S_B of $|C'|$ positions of B, each one associated with a distinct color in C', and a set S_A of $|C'|$ positions of $A[1, i]$, such that there exists a mapping from S_A to S_B that preserves $|C'|$ duos; otherwise, the function is equal to 0.

Define $P[h, i, C']$ as a function equal to 1 when there exist positions q and r in B, with $|C'| = i - h$ and $1 \leq q < r \leq |B|$, such that each color in C' is associated with exactly one position between q and $r - 1$, and substring $B[q, r]$ is identical to $A[h, i]$; otherwise the function is equal to 0.

We can compute $D[i, C']$ as follows:

$$D[i, C'] = \max \begin{cases} \max_{C'' \subseteq C'} D[h, C''] \times P[h+1, i, C' \setminus C''] \\ \qquad\qquad \text{where } h < i - 1,\ i - h - 1 = |C' \setminus C''| \\ D[i-1, C'] \end{cases}$$

In the base case it holds $D[1, C'] = 1$ if $|C'| = 0$, else $D[1, C'] = 0$. There exists a solution of Max-Duo PSM with k preserved duos, if and only if there exists a function $f \in F$ such that $D[n, C] = 1$.

Next we prove the correctness of the recurrence.

Lemma 2. *Given two related strings A and B, there exists a partial mapping of $A[1, i]$ into B that preserves $|C'|$ duos induced by positions of B colored by C' if and only if $D[i, C'] = 1$.*

Proof. We prove the lemma by induction on i. First consider the base case, that is $i = 1$, then $D[1, C'] = 1$ if and only if $|C'| = 0$, since $A[1]$ contains no duo.

Assume that the lemma holds for $j < i$, we show that it holds for $j = i$.

($\Leftarrow$) First, assume that $D[i, C'] = 1$, then we show that there exists a partial mapping of $A[1, i]$ into B that preserves $|C'|$ duos induced by positions of B colored by C'.

By assuming $D[i, C'] = 1$, then, if $D[i-1, C'] = 1$, by induction hypothesis there exists a partial mapping of $A[1, i-1]$ into B that preserves $|C'|$ duos induced by positions of B colored by C'. On the other hand, we have $D[i, C'] = D[h, C''] \times P[h+1, i, C' \setminus C''] = 1$, for some $C'' \subseteq C'$, with $|C' \setminus C''| = i - h - 1$. Then, since $D[h, C''] = 1$, by induction hypothesis there exists a partial mapping of $A[1, h]$ into B that preserves $|C''|$ duos induced by positions of B colored by C''. Moreover, $P[h+1, i, C' \setminus C''] = 1$, and it follows that there exist positions q and r of B such that each color in $C' \setminus C''$ is associated with a distinct position t of B, with $q \leq t \leq r - 1$, and $B[q, r]$ is identical to $A[h+1, i]$. Hence, it follows that $i - h - 1$ duos are preserved by mapping $A[h+1, i]$ into $B[q, r]$, and are induced by positions of B colored by $C' \setminus C''$. As a consequence there exists a partial mapping of $A[1, i]$ into B that preserves $|C'|$ duos induced by positions of B colored by C'.

($\Rightarrow$) Now, assume that there exists a partial mapping of $A[1,i]$ into B that preserves $|C'|$ duos induced by positions of B colored by C'. We show that $D[i,C']=1$. Now, there are two possible cases depending on the fact that position $i-1$ of A induces a preserved duo or not. In the latter case by induction hypothesis, $D[i-1,C']=1$ and hence $D[i,C']=1$.

In the former case, there exists a position h in A, with $1 \leq h \leq i-1$, such that there exists a sequence of preserved consecutive duos $d_{A(h+1,i)}$ mapped into a sequence of preserved consecutive duos $d_{B(z+1,j)}$, with $j-z=i-h$. Since function f assigns a distinct color to each position of B that induces a preserved duo, there exists a set C'' such that each position of $d_{B(z+1,j)}$ inducing a preserved duo with $d_{A(h+1,i)}$ is associated with a distinct color in $C'' \subseteq C'$, and each position of B that induces a preserved duo with a position of $A[1,h]$ is associated with a distinct color of $C' \setminus C''$. Hence, $P[h+1,i,C'']=1$, for some set $C'' \subseteq C'$. Moreover, by induction hypothesis $D[h,C' \setminus C'']=1$, with $i-h-1=|C' \setminus C''|$, and by the first case of the recurrence $D[i,C']=1$. □

From the previous lemma, we can conclude the correctness of the algorithm.

Theorem 1. *Let A and B be two related strings on an alphabet Σ. Then, it is possible to compute if there exists a solution of Max-Duo PSM on instance (A,B,k) in time $(8e)^k\text{poly}(n)$.*

Proof. The correctness of the algorithm follows from the correctness of the dynamic programming recurrence (see Lemma 2). Now, we consider the time complexity of the algorithm. We recall that $n=|A|=|B|$. Assume that there exists a function f in a perfect family F of hash functions, such that f associates a distinct color in C with each position of B that induces a preserved duo.

In order to analyze the time complexity required to compute $D[i,C']$, we first consider the time required to compute $P[h,i,C' \setminus C'']$. $P[h,i,C' \setminus C'']$ can be precomputed in time $O(2^k k^3 n^2)$, before computing the entries of $D[i,C']$. Indeed table $P[h,i,C' \setminus C'']$ contains $O(2^k kn)$ entries, since it holds that (i) $1 \leq i \leq n$, (ii) there exists at most 2^k subsets $C' \setminus C''$, and (iii) $h \geq i-k$. Notice that $h \geq i-k$ is due to the fact that $A[h,i]$ must be identical to a substring $B[q,r]$, where each color of C' is associated with exactly one position of B. Given positions h and i, and subset $C' \setminus C''$, we must check that each color in $C' \setminus C''$ is associated with a position between q and $r-1$ of B, and that substring $B[q,r]$ is identical to $A[h,i]$. This can be done in time $O(k^2 n)$ by checking whether each substring $B[q,r]$ of length bounded by k (there are at most $O(kn)$ of such strings) is identical to $A[h,i]$, and each color in $C' \setminus C''$ is associated with a position j of $B[q,r-1]$.

Now, consider the time complexity to compute $D[i,C']$, once $P[h,i,C' \setminus C'']$ is precomputed. Table $D[i,C']$ contains $O(2^k n)$ entries. Each entry is computed by the recurrence by considering at most $2^k n$ entries. Indeed, in the first case of the recurrence the entries $D[h,C'']$ must be checked, where $h \leq n$ (hence, there are at most n of such values), and $C'' \subseteq C'$ (hence, there are at most 2^k of such subsets). Once h and i are known, $P[h,i,C' \setminus C'']$ can be checked in constant time. It follows that table $D[i,C']$ can be computed in time $O(2^{2k} k^3 n^2)$ (considering the cost to precompute $P[h+1,i,C' \setminus C'']$).

In order to find an injective function f in a perfect family F, we must iterate through the $(2e)^k O(\log(n))$ functions of F. Since the family F can be computed in time $(2e)^k\text{poly}(n)$ [1], and $k \leq n$, it follows that the overall complexity is indeed $(8e)^k\text{poly}(n)$.

□

4. A Reduction to a Polynomial Kernel

In this section, we prove that the Max-Duo PSM problem admits a polynomial size kernel, by presenting a polynomial-time algorithm that, starting from an instance (A,B,k) of Max-Duo PSM, computes an equivalent instance (A',B',k), such that the length of A' and B' is bounded by $O(k^6)$.

The general idea of the reduction is that in *Phase 1*, starting from the related strings A and B, we compute two subsets of duos of A and B, denoted by C_A and C_B respectively, that may eventually be preserved, while any other duo not in these sets will not be preserved. Then, in *Phase 2*, starting from sets C_A and C_B, we compute two related strings A' and B' respectively, so that (A',B',k) is an instance of Max-Duo PSM.

4.1. Phase 1: Constructing Small Sets of Relevant Duos

Here, we present the algorithm that in polynomial-time, starting from the related strings A and B, computes two subsets C_A and C_B, of duos of A and B, respectively, called *candidate sets*, having the following properties:

1. there exists a solution of Max-Duo PSM on instance (A, B, k) if and only if there exist $C'_A \subseteq C_A$, $C'_B \subseteq C_B$, with $|C'_A| = |C'_B| = k$ such that there is a mapping of C'_A into C'_B;
2. C_A and C_B contains $O(k^6)$ duos.

In order to compute C_A and C_B, the algorithm iteratively adds a set of duos of A (of B, respectively) to C_A (C_B, respectively), so that the size of C_A (of C_B, respectively) is bounded by a polynomial function of k. Recall that k denotes the number of duos preserved by a solution of Max-Duo PSM.

Notice that the algorithm is based on three rules that are not data reduction rules, but they add preservable duos to C_A and C_B.

We start by giving the details of our algorithm. First, we consider an easy bound on the length of each sequence of consecutive duos of A and B that can be preserved. Notice indeed that if there exists a sequence of consecutive duos of A having length at least k that can be mapped into a sequence of consecutive duos of B having length at least k (which can be computed in polynomial time), then obviously there exists a solution that preserves at least k duos. Hence, we assume that the following claim holds.

Claim 1. *There is no sequence of consecutive duos of A having length at least k that can be mapped into a sequence of consecutive duos of B having length at least k.*

Now, we are able to define the rules for the Phase 1 of the kernelization (see Fig. 2 for an example illustrating the three rules). The first rule is based on the approach of [4] that leads to a $\frac{1}{4}$-approximation algorithm. The approximation algorithm given in [4] is based on a graph representation of the duos of the given input strings. A maximum matching of this graph is then computed and it is decomposed into four submatchings; the maximum of such submatchings is then returned as the approximated solution of factor $\frac{1}{4}$.

As in [4], we first consider a bipartite graph $G = (V_A \uplus V_B, E)$ associated with the related strings A and B, input of Max-Duo PSM, and defined as follows:

- for each duo in A, there exists a vertex in V_A;
- for each duo in B, there exists a vertex in V_B;
- there exists an edge $\{v_a, v_b\} \in E$ connecting a vertex $v_a \in V_A$ to a vertex $v_b \in V_B$ if and only if they represent a preservable duo.

Now, we are ready to present the first rule of the kernelization algorithm.

Rule 1. Compute (in polynomial time) a maximum matching $M \subseteq E$ of G and define C_A and C_B as the sets of duos corresponding to the endpoints of each edge of M. More precisely

$$C_A = \{v_a \in V_A | \text{there exists } x \in V_B \text{ with } \{v_a, x\} \in M\}$$

and

$$C_B = \{v_b \in V_B | \text{there exists } x \in V_A \text{ with } \{x, v_b\} \in M\}.$$

It can be shown that $|C_A|, |C_B| \leq 4k$, since otherwise we can compute a solution of Max-Duo PSM on instance (A, B, k).

Lemma 3. *Given two related strings A and B, let G be the corresponding graph. Let M be a maximum matching of G and let C_A and C_B be the two sets of duos built by Rule 1. Then, if $|C_A|, |C_B| \geq 4k$, Max-Duo PSM on instance (A, B, k) admits a feasible solution.*

Proof. It is shown in [4] that M can be partitioned in polynomial time into four submatchings M_i, with $1 \leq i \leq 4$, such that each M_i induces a partial mapping of A into B of size $|M_i|$. If there exists a submatching M_i, with $1 \leq i \leq 4$, of size at least k, then there is a partial mapping of A into B which is a solution of Max-Duo PSM on instance (A, B, k).

Now, if $|C_A|, |C_B| \geq 4k$, then there exists a submatching M_i, with $1 \leq i \leq 4$, such that $|M_i| \geq k$. It follows that M_i induces a partial mapping of A into B that preserves at least k duos, hence Max-Duo PSM on instance (A, B, k) admits a feasible solution. □

In the following, we assume that $|M| < 4k$ and $|C_A|, |C_B| < 4k$. Next, we prove another useful property of the computed maximum matching M of G. We denote by M_A (M_B, respectively) the set of vertices of V_A (V_B, respectively) that are endpoints of an edge belonging to M.

Lemma 4. *Consider the symbols $a, b \in \Sigma$ and assume that there exist preservable duos of the related strings A and B, whose corresponding string is $'ab'$. Then, at most one of the sets $V_A \setminus M_A$ and $V_B \setminus M_B$ contains a vertex associated with a duo whose corresponding string is $'ab'$.*

Proof. Assume by contradiction that the lemma does not hold. It follows that there exists one vertex of $v_a \in V_A$ associated with a duo of A whose corresponding string is $'ab'$ and there exists one vertex of $v_b \in V_B$ associated with a duo of B whose corresponding string is $'ab'$, such that v_a, v_b are not endpoints of an edge of M. Hence by adding such an edge to M (which exists by construction of G), it is possible to obtain a matching larger than M, which contradicts the fact that M is maximum. □

We recall that, given two positions $1 \leq i < j \leq n$, $d_{S(i,j)}$ denotes the sequence of *consecutive duos* $(S[i], S[i+1]), \ldots, (S[j-1], S[j])$ and that $d_{S(i-k,i+k)}$ denotes the sequence of duos between position $S[l]$, where $l = \max\{1, i-k\}$, and position $S[r]$, where $r = \min\{n, i+k\}$.

Rule 2. For each duo $(S[i], S[i+1])$ of C_S, with $S \in \{A, B\}$, add to C_S all the duos of $d_{S(i-k,i+k)}$.

Given $S \in \{A, B\}$, we recall that $\bar{S}$ is the string in $\{A, B\} \setminus \{S\}$. It can be shown that the following properties hold.

Lemma 5. *Given a string S, with $S \in \{A, B\}$, consider a duo $(S[i], S[i+1])$ added by Rule 1 to C_S. If there exists a solution of Max-Duo PSM on instance (A, B, k) that maps a sequence $d_{S(i-t_1,i+t_2)}$ with $0 \leq t_1 \leq k$ and $1 \leq t_2 \leq k$, of consecutive duos of S that includes $(S[i], S[i+1])$ into a sequence $d_{\bar{S}(j-u_1,j+u_2)}$ of consecutive duos of $\bar{S}$, then Rule 2 adds all the duos of $d_{S(i-t_1,i+t_2)}$ to C_S.*

Proof. Assume that a solution of Max-Duo PSM on instance (A, B, k) defines a mapping of a sequence $d_{S(i-t_1,i+t_2)}$ of S into a sequence $d_{\bar{S}(j-u_1,j+u_2)}$ of $\bar{S}$. From Claim 1, we can assume that $t_2 + t_1 + 1 = u_2 + u_1 + 1 \leq k$, thus $t_1, t_2 \leq k$ and $u_1, u_2 \leq k$. Since Rule 2 adds to C_S the sequence $d_{S(i-k,i+k)}$ of duos, the lemma holds. □

Moreover, we can bound the number of duos added by Rule 2 as follows.

Lemma 6. *Rule 2 adds at most $8k^2$ duos to each set C_S, with $S \in \{A, B\}$.*

Proof. Since $|M| < 4k$, it follows that there exist less than $4k$ positions i, with $1 \leq i \leq n$, such that the sequence $d_{S(i-k,i+k)}$ of duos are added to C_S. Since there exist $2k$ consecutive duos in $d_{S(i-k,i+k)}$, at most $8k^2$ duos are added to C_S. □

We are now able to define Rule 3.

Rule 3. Consider a sequence $d_{S(p,q)}$ of consecutive duos that has length at most k, such that each duo $(S[i], S[i+1])$, with $p \leq i \leq q-1$, is added by Rule 1 and Rule 2 to set C_S; add a set of candidate duos to $C_{\bar{S}}$ as follows:

- if there exist at least k^2+1 non-overlapping sequences in $\bar{S}$ where $d_{S(p,q)}$ can be mapped: add to $C_{\bar{S}}$ all the duos belonging to the leftmost non-overlapping k^2+1 sequences in $\bar{S}$ where $d_{S(p,q)}$ can be mapped;
- else, add to $C_{\bar{S}}$ all the duos belonging to the sequences of consecutive duos in $\bar{S}$ where $d_{S(p,q)}$ can be mapped.

Lemma 7. *Consider a solution X of Max-Duo PSM on instance (A, B, k). Then, there exist subsets of duos $C'_A \subseteq C_A$ and $C'_B \subseteq C_B$, such that there is a mapping of C'_A into C'_B and $|C'_A| = |C'_B| = k$.*

Proof. Consider a duo $(S[i], S[i+1])$, where $S \in \{A, B\}$, $S[i] = a$ and $S[i+1] = b$, such that a solution X of Max-Duo PSM on instance (A, B, k) maps a sequence $d_{S(p,q)}$ of consecutive duos of S, with $i-k \leq p < q \leq i+k$, into a sequence of consecutive duos $d_{\bar{S}(t,u)}$. Notice that all the duos of $d_{S(p,q)}$ belong to C_S, or all the duos of $d_{\bar{S}(t,u)}$ belong to $\bar{S}$. Indeed, assume that the duo $(S[i], S[i+1])$ is mapped into the duo $(\bar{S}[j], \bar{S}[j+1])$ by X. By Lemma 4, it follows that Rule 1 adds duo $(S[i], S[i+1])$ to C_S or $(\bar{S}[j+1], \bar{S}[j+1+1])$ to $C_{\bar{S}}$. Moreover, by Lemma 5, Rule 2 adds all the duos of $d_{S(p,q)}$ to C_S or all the duos of $d_{\bar{S}(t,u)}$ to $\bar{S}$. In what follows, we assume w.l.o.g. that all the duos $d_{S(p,q)}$ belong to C_S.

Let S be the string having the maximum number of occurrences of duos whose corresponding string is $'ab'$. Then, by Lemma 4 and by Rule 1, each duo of $\bar{S}$, whose corresponding string is $'ab'$, is added to $C_{\bar{S}}$. Moreover, consider a sequence $d_{S(p,q)}$ of consecutive duos of S, with $i-k \leq p < q \leq i+k$, mapped by X into a sequence $d_{\bar{S}(t,u)}$ of consecutive duos of $\bar{S}$. By Lemma 5, all the duos of $d_{\bar{S}(t,u)}$ are added to $C_{\bar{S}}$ by Rule 2. Hence, we define $C'_S \subseteq C_S$ and $C'_{\bar{S}} \subseteq C_{\bar{S}}$ so that they include the duos of $d_{S(p,q)}$ and the duos of $d_{\bar{S}(t,u)}$, respectively, and there is a mapping between such duos.

Assume that S is the string having the minimum number of occurrences of $'ab'$ and consider the sequence $d_{S(p,q)}$ of consecutive duos of X mapped into the sequence $d_{\bar{S}(t,u)}$ of consecutive duos. By Rule 3, either all the consecutive duos of $\bar{S}$ where $d_{S(p,q)}$ can be mapped are included in $C_{\bar{S}}$, or the duos of k^2+1 non-overlapping sequences of consecutive duos of $\bar{S}$, where $d_{S(p,q)}$ can be mapped, are included in $C_{\bar{S}}$.

In the former case, all the duos of $d_{\bar{S}(t,u)}$ are added to $C_{\bar{S}}$ by Rule 3, hence the duos of $d_{S(p,q)}$ and the duos of $d_{\bar{S}(t,u)}$ are included in C'_S and $C'_{\bar{S}}$ respectively, and there is a mapping between such duos.

In the latter case, we show that, even if $C_{\bar{S}}$ does not contain all the duos of $d_{\bar{S}(t,u)}$, it contains a sequence of consecutive duos where $d_{S(p,q)}$ can be mapped. Notice that each sequence of consecutive duos of $\bar{S}$ mapped to sequence of consecutive duos of S has length bounded by k (see Claim 1) and, since X preserves k duos, the set $C'_{\bar{S}}$ contains at most k such sequences of consecutive duos. It follows that each sequence of consecutive duos added to $C'_{\bar{S}}$ can overlap at most k non-overlapping occurrences of $d_{S(p,q)}$ in $\bar{S}$. Hence the whole set of sequences of consecutive duos preserved of $C'_{\bar{S}}$ can overlap at most k^2 non-overlapping sequence of consecutive duos in $\bar{S}$ where $d_{S(p,q)}$ can be mapped. Since we have added to $C_{\bar{S}}$ the duos belonging to k^2+1 non-overlapping sequences of consecutive duos of $\bar{S}$ where $d_{S(p,q)}$ can be mapped, then there exists at least a sequence $d_{\bar{S}(w,y)}$ of consecutive duos in $C_{\bar{S}}$ where $d_{S(p,q)}$ can be mapped. Hence, the duos of $d_{S(p,q)}$ and the duos of $d_{\bar{S}(w,y)}$ are included in C'_S and $C'_{\bar{S}}$ respectively, and there is mapping between such duos. □

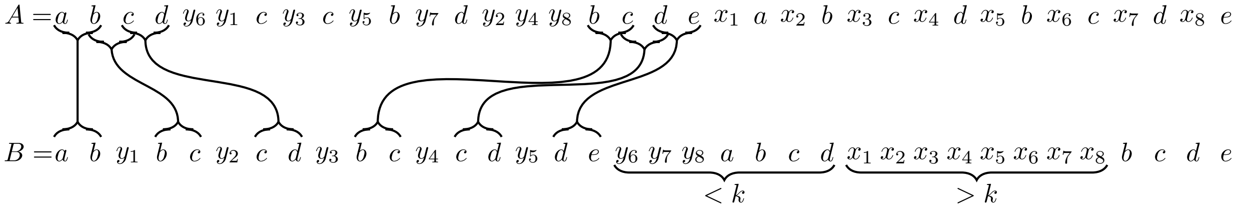
Now, prove a bound on the size of the sets C_A and C_B.

Lemma 8. *Given an input string $S \in \{A, B\}$, Rules 1-3 add at most $O(k^6)$ duos to each set C_S.*

Proof. By Lemma 4, Rule 1 adds at most $4k$ duos to each set C_S, with $S \in \{A, B\}$. By Lemma 6, Rule 2 adds at most $O(k^2)$ duos to each set C_S, with $S \in \{A, B\}$.

Rule 3 considers $O(k^3)$ sequences $d_{S(p,q)}$ of length bounded by k. For each such sequence of consecutive duos, Rule 3 adds duos to $C_{\bar{S}}$ in two possible ways. In the first case k^2+1 sequences of consecutive duos are selected, each one having size at most k, thus $O(k^3)$ duos are added to $C_{\bar{S}}$; thus $O(k^6)$ are added to $C_{\bar{S}}$.

In the second case, for each sequence $d_{S(p,q)}$, at most k^3 duos are added to $C_{\bar{S}}$. Indeed, Rule 3 adds all the duos belonging to sequences of consecutive duos in $\bar{S}$ where $d_{S(p,q)}$ can be mapped. There exist at most



A = a b c d y6 y1 c y3 c y5 b y7 d y2 y4 y8 b c d e x1 a x2 b x3 c x4 d x5 b x6 c x7 d x8 e
B = a b y1 b c y2 c d y3 b c y4 c d y5 d e y6 y7 y8 a b c d x1 x2 x3 x4 x5 x6 x7 x8 b c d e
< k
> k

Figure 2: Two related strings A and B, where two duos are connected by a line if the corresponding edge belongs to a given maximum matching. Phase 1 of the algorithm adds the duos connected by a line to C_A and C_B. Phase 2 adds the duos induced by *abcd* of B since it belongs to a substring $d_{S(i-k,i+k)}$. Phase 3 adds the duos induced by *bcde* of B as there is only one occurrence (hence $< k$) of *bcde* in B. Notice that a solution of Max-Duo PSM that preserves 6 duos exists only if the sequence of duos *abcd* and *bcde* of A are mapped to the sequences of duos *abcd* and *bcde* of B.

k^2 non-overlapping sequences of consecutive duos (each one having length at most k) where $d_{S(p,q)}$ can be mapped and, writing s' for one of these sequences of consecutive duos, s' can overlap at most $2k$ duos where $d_{S(p,q)}$ can be mapped (at most k duos on the left and k duos on the right of s').

Hence, the overall number of consecutive duos added to $C_{\bar{S}}$ is $O(k^6)$. □

4.2. Phase 2: Completing the construction

From the sets C_A and C_B previously computed, we construct an instance of Max-Duo PSM, that is two related strings A' and B'. Furthermore, we will show that the length of A' and B' is bounded by $O(k^6)$ and that the preservable duos of A' and B' are those of C_A and C_B, respectively.

Recall that A and B are two related strings over alphabet Σ. Consider the set T_A of substrings of A (the set T_B of substrings of B, respectively) that induces the duos in C_A (in C_B, respectively) and assume that T_S, with $S \in \{A, B\}$, contains q_S strings, namely $t_{1,S}, \dots t_{q_S,S}$.

Before describing the two strings A' and B', we construct the alphabet Σ' on which they are based, where:

$$\begin{aligned}\Sigma' = \ & \Sigma \ \cup \\ & \{e_{A,i} : 1 \leq i \leq q_A\} \ \cup \\ & \{e_{B,i} : 1 \leq i \leq q_B\} \ \cup \\ & \{g_a : a \in \Sigma \text{ has a different number of occurrences in } T_A \text{ and } T_B\}\end{aligned}$$

We concatenate the substrings of T_A (of T_B, respectively) with symbols of $\Sigma' \setminus \Sigma$. We compute an intermediate string P_S, with $S \in \{A, B\}$, as follows. First, set $P_S = t_{1,S} \cdot e_{S,1}$ (that is, the concatenation of $t_{1,S}$ and $e_{S,1}$). Then, for each i with $2 \leq i \leq q_S$, append the string $t_{i+1,S} \cdot e_{S,i}$ to the string P_S. Finally, append string $e_{\bar{S},1} \dots e_{\bar{S},q_S}$ at the right end of P_S.

Now, we append some strings to the right end of P_A and P_B in order to compute A' and B' over alphabet Σ'. More precisely, for each symbol $a \in \Sigma$ such that the number of occurrences of a in P_A and in P_B is different, we apply the following procedure. Assume w.l.o.g. that P_A contains $h_{A,a}$ occurrences of symbol a and P_B contains $h_{B,a}$ occurrences of symbol a, with $h_{A,a} > h_{B,a}$. Then, we append a string $(g_a a)^{h_{A,a}-h_{B,a}}$ (that is the string consisting of the concatenation of $h_{A,a} - h_{B,a}$ occurrences of $g_a a$) to the right end of P_B, while we append the string $(g_a)^{h_{A,a}-h_{B,a}}$ (that is the string consisting of the concatenation of $h_{A,a} - h_{B,a}$ occurrences of g_a) to the right end of P_A. Similarly, if $h_{A,a} < h_{B,a}$, we append a string $(g_a a)^{h_{B,a}-h_{A,a}}$ to the right end of P_A, while we append the string $(g_a)^{h_{B,a}-h_{A,a}}$ to the right end of P_B (see Figure 3).

The following lemma guarantees that the instance built on A' and B', is an instance of Max-Duo PSM.

Lemma 9. *Let A' and B' be the two strings computed starting from T_A and T_B, respectively. Then, A' and B' are related. Moreover, $|A'|$ and $|B'|$ are bounded by $O(k^6)$.*

Proof. Notice that each symbol in $\Sigma' \setminus \Sigma$ by construction has the same number of occurrences in A' and B'. Moreover, for each symbol in $a \in \Sigma$, both A' and B' contain by construction $\max(h_{A,a}, h_{B,a})$ occurrences of a, hence A' and B' are related.

$$\begin{aligned} A' &= \overbrace{t_{1,A} \cdot e_{A,1} \dots t_{q_A,A} \cdot e_{A,q_A}}^{P_A} e_{B,1} \dots e_{B,q_B} \; \overbrace{g_a g_a \cdots g_a}^{(g_a)^{h_{A,a}-h_{B,a}}} \; \overbrace{g_b g_b \cdots g_b}^{(g_b)^{h_{A,b}-h_{B,b}}} \; \overbrace{g_c c g_c c \dots g_c c}^{(g_c c)^{h_{B,c}-h_{A,c}}} \\ B' &= \underbrace{t_{1,B} \cdot e_{B,1} \dots t_{q_B,B} \cdot e_{B,q_B}}_{P_B} e_{A,1} \dots e_{A,q_A} \underbrace{g_a a g_a a \dots g_a a}_{(g_a a)^{h_{A,a}-h_{B,a}}} \underbrace{g_b b g_b b \dots g_b b}_{(g_b b)^{h_{A,b}-h_{B,b}}} \; \underbrace{g_c g_c \cdots g_c}_{(g_c)^{h_{B,c}-h_{A,c}}} \end{aligned}$$

Figure 3: An example of two related strings A' and B' constructed during Phase 2 from sets C_A and C_B (and sets $T_A$and T_B of substrings), respectively. Notice that the string T_A contains $h_{A,a}$, $h_{A,b}$, and $h_{A,c}$ occurrences of symbols a, b, and c, respectively, while the string T_B contains $h_{B,a}$, $h_{B,b}$, and $h_{B,c}$ occurrences of symbols a, b, and c, respectively. Moreover, we assume that $h_{A,a} > h_{B,a}$, $h_{A,b} > h_{B,b}$, and $h_{A,c} < h_{B,c}$. Hence, strings $g_a a \dots g_a a$ and $g_b b \dots g_b b$ are appended to construct A', while strings $g_a \dots g_a$ and $g_b \dots g_b$ are appended to construct B'; string $g_c c \dots g_c c$ is appended to construct B', while string $g_c \dots g_c c$ is appended to construct A'.

From Lemma 8 it follows that $|C_A|$ and $|C_B|$ are bounded by $O(k^6)$, hence the symbols $e_{S,i}$, with $1 \leq i \leq q_S$ (where $S \in \{A, B\}$), inserted in A' and B' are at most $O(k^6)$. Similarly, the number of symbols $g_a \notin \Sigma$ and $a \in \Sigma$ inserted are at most $O(k^6)$, since in the worst case at most $|C_A|$ ($|C_B|$, respectively) symbols of Σ are inserted in A' (in B', respectively), and for each of such symbols, exactly one occurrence of some symbol g_a is inserted in A' and B'. □

Finally, the following property holds.

Lemma 10. *Let X be a solution of Max-Duo PSM on instance (A', B', k), such that X maps $(A'[i], A'[i+1])$ into $(B'[j], B'[j+1])$. Then $(A'[i], A'[i+1])$ and $(B'[j], B'[j+1])$ are duos of C_A and C_B, respectively.*

Proof. The lemma follows from the fact that, by construction, each preservable duo of A' and B' belongs to strings in T_A and T_B, respectively. Indeed, consider w.l.o.g. a duo $(S'[i], S'[i+1])$ of S', with $S' \in \{A', B'\}$, not in C_S. Then, by construction of S', $(S'[i], S'[i+1])$ must include at least a symbol x in $\Sigma' \setminus \Sigma$, as each new occurrence of a symbol in Σ is adjacent to a symbol in $\Sigma' \setminus \Sigma$. Now, notice that if x is equal to e_{S,j_S}, with $1 \leq j \leq q_S$, then by construction it belongs to duos with different symbols in S' and $\bar{S}'$, as x is adjacent in $\bar{S}'$ only to symbol in $\Sigma' \setminus \Sigma$, while in S is adjacent only to symbols in Σ, with the exception of e_{S,q_S}, which again by construction cannot belong to a preservable duo.

Now, consider a symbol $g_y \in \Sigma' \setminus \Sigma$. Then exactly one of A', B' contains a duo whose corresponding string is yg_y and exactly one of A', B' contains a duo whose corresponding string is $g_y g_y$. Moreover, exactly one of A', B' contains a duo whose corresponding string is zg_y and exactly one of A', B' contains a duo whose corresponding string is $g_z g_y$, with z a symbol in Σ, with the exception of the first symbol $g_a \in \Sigma' \setminus \Sigma$ appended to P_A or P_B, which again by construction cannot belong to a preservable duo. □

We conclude the description of the kernelization algorithm with the following theorem.

Theorem 2. *Given an instance (A, B, k) of Max-Duo PSM, Phase 1 and Phase 2 compute in time $O(n^{\frac{5}{2}} k^6)$ an instance (A', B', k), with $|A'|$ and $|B'|$ bounded by $O(k^6)$, such that there exists a solution of Max-Duo PSM on instance (A, B, k) if and only if there exists a solution of Max-Duo PSM on instance (A', B', k).*

Proof. Notice that by Lemma 9, $|A'|$ and $|B'|$ are bounded by $O(k^6)$. First, we show that there exists a solution of Max-Duo PSM on instance (A, B, k) if and only if there exists a solution of Max-Duo PSM on instance (A', B', k).

($\Rightarrow$) Consider a solution of Max-Duo PSM on instance (A', B', k), then a solution Max-Duo PSM on instance (A, B, k) can be computed by preserving those duos of (A, B, k) corresponding to the duos preserved by the solution of Max-Duo PSM on instance (A', B', k).

($\Leftarrow$) By Lemma 7 and by Lemma 10, if there exists a solution of of Max-Duo PSM on instance (A, B, k), then there exists a solution of of Max-Duo PSM on instance (A', B', k).

Now, we show that Phase 1 and Phase 2 compute instance (A', B', k) in time $O(n^{\frac{5}{2}} k^6)$. Rule 1 requires time $O(n^2)$ to compute the graph G and $O(n^{\frac{5}{2}})$ to compute a maximum bipartite matching of G [16]. Rule

2 requires time $O(k^2)$ to add to C_A and C_B, for each of the at most $O(k)$ duos of M_A and M_B, the $O(k)$ duos of $d_{S(i-k,i+k)}$. Rule 3 requires time $O(k^6 n)$ to add to C_A and C_B, since it considers $O(k^3)$ sequences $d_{S(p,q)}$ for which $O(k^2)$ sequences $d_{\bar{S}(r,t)}$ of length k. Moreover, time $O(n)$ is required to find the occurrences of each sequence $d_{S(p,q)}$ in $\bar{S}$.

Finally, Phase 2 computes A' and B' in time $O(k^6)$. Indeed, P_S can be computed in time $O(k^6)$ by concatenating a set of $O(k^6)$ strings. Moreover, we can count the number of occurrences of symbols of Σ in a string $P_S \in \{A, B\}$ in time $O(k^6)$, and we append $O(k^6)$ strings $g_a a$, g_a in time $O(k^6)$. □

5. Conclusion

In this paper, we have investigated the parameterized complexity of the Max-Duo PSM problem, by first giving a parameterized algorithm based on color-coding and then showing that it admits a kernel of size $O(k^6)$. Notice that the kernel is obtained by extending the alphabet Σ, in order to ensure that all the preservable duos of strings A' and B' belong to sets C_A and C_B. If we don't extend the alphabet, we may introduce some new duo that do not belong to the solution of Max-Duo PSM over instance (A, B, k). Notice, however, that the sets C_A, C_B, and the kernel have size at most $O(k^6)$.

From a paramterized complexity point of view, there are some interesting open problems for Max-Duo PSM. First, following the approach of *parameterizing above a guaranteed value*, it would be interesting to investigate the parameterized complexity of the problem when the parameter is the number of conserved duos minus the conserved duos induced by the submatching returned by the approximation algorithm in [4]. Furthermore, it would be interesting to improve upon the time (and space) complexity of our color-coding based algorithm. In particular, notice that our color-coding based algorithm requires exponential space complexity, as it makes use of two tables of size $O(2^k kn)$.

Acknowledgments

We would like to thank the anonymous reviewers for very helpful and constructive comments that led to a significant improvement of the paper.

References

[1] Alon, N., Yuster, R., Zwick, U.: Color-coding. Journal of the ACM 42(4), 844–856 (1995)

[2] Bonizzoni, P., Della Vedova, G., Dondi, R., Pirola, Y.: Variants of constrained longest common subsequence. Inf. Process. Lett. 110(20), 877–881 (2010)

[3] Bonizzoni, P., Dondi, R., Mauri, G., Zoppis, I.: Restricted and swap common superstring: A multivariate algorithmic perspective. Algorithmica 72(4), 914–939 (2015), http://dx.doi.org/10.1007/s00453-014-9882-8

[4] Boria, N., Kurpisz, A., Leppänen, S., Mastrolilli, M.: Improved approximation for the maximum duo-preservation string mapping problem. In: Brown, D., Morgenstern, B. (eds.) Algorithms in Bioinformatics - 14th International Workshop, WABI 2014, Wroclaw, Poland, September 8-10, 2014. Proceedings. Lecture Notes in Computer Science, vol. 8701, pp. 14–25. Springer (2014)

[5] Bulteau, L., Carrieri, A.P., Dondi, R.: Fixed-parameter algorithms for scaffold filling. Theor. Comput. Sci. 568, 72–83 (2015), http://dx.doi.org/10.1016/j.tcs.2014.12.005

[6] Bulteau, L., Fertin, G., Komusiewicz, C., Rusu, I.: A fixed-parameter algorithm for minimum common string partition with few duplications. In: WABI. pp. 244–258 (2013)

[7] Bulteau, L., Komusiewicz, C.: Minimum common string partition parameterized by partition size is fixed-parameter tractable. In: Chekuri, C. (ed.) Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2014, Portland, Oregon, USA, January 5-7, 2014. pp. 102–121. SIAM (2014), http://dx.doi.org/10.1137/1.9781611973402.8

[8] Chen, W., Chen, Z., Samatova, N.F., Peng, L., Wang, J., Tang, M.: Solving the maximum duo-preservation string mapping problem with linear programming. Theor. Comput. Sci. 530, 1–11 (2014), http://dx.doi.org/10.1016/j.tcs.2014.02.017

[9] Chen, X., Zheng, J., Fu, Z., Nan, P., Zhong, Y., Lonardi, S., Jiang, T.: Assignment of orthologous genes via genome rearrangement. IEEE/ACM Trans. Comput. Biology Bioinform. 2(4), 302–315 (2005), http://doi.acm.org/10.1145/1100863.1100950

[10] Chrobak, M., Kolman, P., Sgall, J.: The greedy algorithm for the minimum common string partition problem. ACM Transactions on Algorithms 1(2), 350–366 (2005), http://doi.acm.org/10.1145/1103963.1103971

[11] Cormode, G., Muthukrishnan, S.: The string edit distance matching problem with moves. ACM Transactions on Algorithms 3(1) (2007), http://doi.acm.org/10.1145/1219944.1219947

[12] Damaschke, P.: Minimum common string partition parameterized. In: Crandall, K.A., Lagergren, J. (eds.) Algorithms in Bioinformatics, 8th International Workshop, WABI 2008, Karlsruhe, Germany, September 15-19, 2008. Proceedings. Lecture Notes in Computer Science, vol. 5251, pp. 87–98. Springer (2008)

[13] Downey, R., Fellows, M.: Fundamentals of Parameterized Complexity. Springer (2013)

[14] Fellows, M.R., Fertin, G., Hermelin, D., Vialette, S.: Upper and lower bounds for finding connected motifs in vertex-colored graphs. J. Comput. Syst. Sci. 77(4), 799–811 (2011), http://dx.doi.org/10.1016/j.jcss.2010.07.003

[15] Goldstein, A., Kolman, P., Zheng, J.: Minimum common string partition problem: Hardness and approximations. Electr. J. Comb. 12 (2005)

[16] Hopcroft, J.E., Karp, R.M.: An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs. SIAM J. Comput. 2(4), 225–231 (1973)

[17] Jiang, H., Zhu, B., Zhu, D., Zhu, H.: Minimum common string partition revisited. J. Comb. Optim. 23(4), 519–527 (2012), http://dx.doi.org/10.1007/s10878-010-9370-2

[18] Kolman, P., Walen, T.: Reversal distance for strings with duplicates: Linear time approximation using hitting set. Electr. J. Comb. 14(1) (2007)

[19] Niedermeier, R.: Invitation to Fixed-Parameter Algorithms. Oxford University Press (2006)